# Digital Fragile Watermarking Scheme For Authentication Of JPEG Images

*Chang-Tsun Li*

Department of Computer Science, University of Warwick,
Coventry CV4 7AL, UK
ctli@dcs.warwick.ac.uk

**Abstract**

It is a common practice in transform-domain fragile watermarking schemes for authentication purposes to watermark some selected transform coefficients so as to minimise embedding distortion. However, we point out in this work that leaving most of the coefficients unmarked results in a wide-open security gap for attacks to be mounted on them. A fragile watermarking scheme is proposed to *implicitly* watermark all the coefficients by *registering* the zero-valued coefficients with a key-generated binary sequence to create the watermark and involving the unwatermarkable coefficients during the embedding process of the embeddable ones. Non-deterministic dependence is established by involving some of the unwatermarkable coefficients selected according to the watermark from a *9-neighbourhood system* in order to thwart different attacks such as cover-up, vector quantisation, and transplantation. No hashing and cryptography are needed in establishing the non-deterministic dependence.

**Keywords:** fragile watermarking, image authentication, integrity verification, multimedia security

## 1. INTRODUCTION

In response to the threat of forgery posed by the power of multimedia processing tools

and the prevalence of interconnected networks, researchers have been actively investigating methods of information hiding for authenticating and verifying the content integrity of the multimedia in the last decade. Various types of fragile watermarking schemes [1, 2, 5, 8-10, 15-20] have been proposed to serve these purposes. The watermarks of the *robust* watermarking schemes [7, 12] for copyright protection are expected to *survive* different types of manipulations to some extent provided that the manipulated media are still valuable in terms of commercial importance or significant in terms of visual quality. Unlike robust schemes, the schemes for the purposes of authentication and content integrity verification are supposed to be *fragile*, i.e., we expect the watermark to be destroyed when attacks are mounted on its host media so that alarms can be raised when wrong watermark is extracted. Therefore, the emphasis of the fragile watermarking schemes is focused on the sensitivity to attacks [11, 19] or even incidental manipulations in some cases [1, 8, 9, 20].

To be considered effective, a fragile watermarking scheme must meet the common requirements such as localising tampering, detecting geometric transformations (e.g., cropping and scaling), signalling removal of original objects, addition of foreign objects, and alerting other image processing operations (e.g., low-pass filtering). In addition, it is more pragmatic to authenticate the media without referring to the original unwatermarked version. This feature is commonly referred to as blind detection [3].

Moreover, it is equally important that a fragile watermarking scheme must show no security gaps to attacks such as cover-up / cut-and-paste [1] and vector quantization [17] (also known as birthday attack [1], the Holliman-Memon counterfeiting attack [6], or collage attack [4, 13]). Cover-up attack is the action of cutting one region / block of the

image and pasting it somewhere in the same or another image. Vector quantization / birthday attacks are devised on the basis of the so-called *birthday paradox* [14, Appendix 8.A]: *What is the minimum population size such that the probability that at least two of the people have the same birthday is greater than 0.5?* According to birthday paradox, using a hash function that produces a bit string of length $l$, the probability of finding at least two blocks that hash to the same output is greater than 0.5 whenever roughly $2^{l/2}$ watermarked blocks are available. The idea of the attack is to forge a new watermarked image (a collage) from a number of authenticated images watermarked with the same key and the same logo / watermark by combining portions of different authenticated images while preserving their relative positions in the image. Fridrich *et al.* [4] showed that counterfeiting is possible even when the logo is unknown to the attacker provided that a larger number of images watermarked with the same key are available.

Block-wise dependence is accepted as an essential requirement to combat vector quantization / birthday attacks [4-6, 8-10, 17]. However, it has also been shown that dependence with deterministic context, i.e., the information involved or dependent upon is deterministic, is susceptible to transplantation attacks or even simple cover-up attacks [1]. The 'transplantation attack' derived by Barreto *et al*. [1] works as follows. For example, let $I'_A \rightarrow I'_B$ denote that the hashing of block $I'_B$ involves the information about $I'_A$. Now, if images $I'$ and $I''$ have blocks with following dependence relationships:

$$\ldots \rightarrow I'_A \rightarrow I'_X \rightarrow I'_B \rightarrow I'_C \rightarrow \ldots$$

$$\ldots \rightarrow I''_A \rightarrow I''_X \rightarrow I''_B \rightarrow I''_C \rightarrow \ldots$$

and block $I'_A$ is identical to $I''_A$, $I'_B$ is identical to $I''_B$, and $I'_C$ is identical to $I''_C$, but $I'_X$ is not identical to $I''_X$. Then the positions of block pairs ($I'_X$, $I'_B$) and ($I''_X$, $I''_B$) are interchangeable

without being detected by schemes adopting deterministic dependence [8, 16, 17, 20]. Barreto *et al.* further indicated that merely increasing the number of dependences could not thwart the transplantation attack. For example, let $I_A \leftrightarrow I_B$ denote that the hashing of each block involves the information about the other. Now if the following dependence relationships exist

$$\cdots \leftrightarrow I'_A \leftrightarrow I'_B \leftrightarrow I'_X \leftrightarrow I'_C \leftrightarrow I'_D \leftrightarrow \cdots$$

$$\cdots \leftrightarrow I''_A \leftrightarrow I''_B \leftrightarrow I''_X \leftrightarrow I''_C \leftrightarrow I''_D \leftrightarrow \cdots,$$

the triplet ($I'_B$ ,$I'_X$ , $I'_C$) and ($I''_B$ ,$I''_X$ , $I''_C$) are interchangeable if block $I'_D$ is also identical to $I''_D$.

Usually, spatial-domain fragile watermarking schemes [1, 8-10, 16, 20] watermark all the pixels. However, they are not directly suitable for some applications where transformation is needed to compress the images. For example, JPEG is one of the most popular standards for transmitting and storing images in compressed format in order to make efficient use of bandwidth and storage. Although some transform-domain schemes have been proposed to meet this requirement, we observed that many of the schemes [15, 18, 19] watermarked only some selected coefficients while leaving most coefficients unprotected in order to minimize the embedding distortion. As a result, a wide security gap is left open to attacks. Our observation suggests that measures of protecting *all* the coefficients without actually watermarking all coefficients and compromising the visual quality of the image are desirable. Due to the fact that JPEG is one of the most common standards for image storage and transmission over the computer networks, our intention in this work is to propose a transform-domain fragile watermarking scheme, which meets the afore-mentioned requirements, for authentication and content integrity verification of

JPEG images.

The rest of this work is organised as follows. Sec. 2 reviews some related works and discusses their merits and limitations. Sec. 3 proposes and analyses the new scheme. Experiments are conducted in Sec. 4 to test the proposed scheme. Finally, Sec. 5 concludes this work.

## 2. RELATED WORKDS AND THEIR LIMITATIONS

In Wong's public-key scheme [16], the LSB-zeroed target image and the binary watermark image are divided into blocks of the same size. The image size together with each LSB-zeroed image block is then provided as inputs to a hash function and the output together with the watermark block are subjected to an exclusive-or (XOR) operation. The result of the XOR operation is then encrypted using a private key and embedded in the least significant bits of the original image. This scheme marries cryptography and watermarking elegantly and indeed works well in detecting cropping, and scaling. However, due to the lack of mutual dependence among neighbouring blocks during the watermarking process, this scheme is vulnerable to cover-up, vector quantisation, and transplantation attacks.

Wu and Liu proposed a scheme [18] that inserts a binary watermark sequence into the DCT coefficients via a look-up table, which maps all possible values of DCT coefficients randomly to either 1 or 0. In the embedding process, to embed 1 in a DCT coefficient, the coefficient is kept unchanged if the corresponding entry of that coefficient is also 1 in the look-up table. If the corresponding entry is 0, the coefficient is changed to the closest value whose entry is 1 in the look-up table. A similar process is applied to the case for embedding 0s. At the receiver side, the extraction of the watermark is simply done by looking up the

table. Although, this scheme does not require the original image for watermark extraction, the same look-up table used in the embedding stage is necessary in the watermark extraction stage, which has to be transmitted through a secure channel and may compromise the security of the scheme. Moreover, like Wong's scheme [16], this scheme is also block-wise independent, and, therefore, vulnerable to cover-up, vector quantisation, and transplantation attacks.

Recognizing the importance of establishing dependence among neighboring pixels or blocks, we proposed a scheme [8] that uses a binary feature map extracted from the underlying image as watermark. The watermark is then divided into blocks of size $32 \times 16$ pixels. Block-wise dependence is established by blending the neighbouring blocks before encrypting and embedding into LSBs of the image. This method is effectively resistant to vector quantization and cover-up attacks and requires no *a priori* knowledge about the image to be watermarked. However, the accuracy of localization is limited by the block size. Moreover, like schemes of [16] and [18], this scheme is also vulnerable to transplantation attack because the contextual dependence is established based on deterministic information. To circumvent these drawbacks, we further proposed a scheme [9, 10], which is immune to transplantation attack and is significantly accurate in locating tampering. However, it is a spatial-domain approach, which is not suitable for transform-domain applications.

To thwart transplantation attack, Barreto *et al.* [1], who pointed out how the attack is possible, proposed to generate the fingerprint of each image block through the calculation of a hash function taking the target block, a neighbouring block, and some random data as inputs. Since the random data is unique to each block, the fingerprint is thus unique and

non-deterministic. However, the hash operation is relatively time-consuming and the accuracy of the tampering localization is limited by the size of the block. Moreover, as a spatial domain approach, this scheme is not suitable for transform-domain applications either.

Although there are some transform-domain schemes reported in the literature, a common security gaps inherent in many of them [15, 18, 19] is that they neither *explicitly* nor *implicitly* watermark all the transform coefficients. As a result, manipulation of those unwatermarked coefficients will go unnoticed. For example, in the wavelet transform-domain approach proposed by Winne *at el*. [15], to minimize the embedding distortion and maintain high localization accuracy, only the coefficients of the high frequency sub-bands at the finest scale of the luminance component are watermarked. All the other coefficients and components are neither watermarked nor involved during the watermarking process of the embeddable coefficients. In [19], to make the scheme semi-fragile, only the LL component of the coarsest scale are involved in generating the signature, which is then used as the watermark. To minimize embedding distortion, only the coefficients of the finest scale are watermarked. Consequently, tampering the coefficients in other sub-bands and scales will certainly go undetected. For example, locally tampering the three unwatermarked high frequency sub-bands at the coarsest scale, which are not involved in generating the signature, is highly likely to change or at least destroy the semantic meaning of the watermarked image without raising alarm.

In Fridrich *et al*'s work [5], all the coefficients are protected by taking all quantised DCT coefficients as input to the hash function and using the hash output as the signature, which is then embedded in the least significant bits of the lossless compressed version of some

selected coefficients. However, the hash output conveys only global information about the image. When a local attack is launched against the coefficients that are not selected for embedding the hash, their algorithm can only tell that the image is not authentic without being able to locate the position where the tampering occurs.

In the light of the limitations of the reviewed schemes, it is desirable to design a transform-domain scheme, which is immune to the afore-mentioned attacks and provides protection for *all* the transform coefficients without explicitly watermarking all of them. Given the popularity of the JPEG standard, which adopts Discrete Cosine Transform (DCT), a scheme watermarking the DCT coefficients is proposed in the next section for authenticating JPEG images.

## 3. PROPOSED WATERMARKING SCHEME

To embed the watermark, the target image is first DCT transformed and quantised. A binary sequence, *A,* with the length of the same size as the image, is generated with a secret key. A second binary map, *B*, is then created so that all its pixels corresponding to the non-zero-valued coefficients are set to 1 and the others set to 0. *B* is intended to serve the purpose of registering the positions of the zero-valued coefficients. A binary watermark, *W*, is then created by taking the result of EXCLUSIVE-OR operation on the binary sequences *A* and *B*. For each DCT block, four non-zero coefficients with their frequencies lower or equal to a middle frequency *h* are identified as watermarkable. The four selected coefficients are modulated based on their corresponding watermark bits in *W* and a secret sum calculated by adding up the non-zero coefficients picked from a neighbourhood system according to their corresponding watermark bits in *W*. The watermarking process

repeats until all the blocks are marked. To authenticate and verify the received image, the verifier performs the same operations as applied on the embedding side in the reversed order to extract the embedded watermark and compares it with the original watermark generated in the same manner as that adopted by the embedder.

In order to make the algorithm clearer, some symbols are defined and explained as follows before the presentation of the algorithms.

$X$: the set of quantised DCT coefficient blocks of the original image. It can be represented as $X = \{X_0, X_1, X_2, ..., X_i, ..., X_{p-1}\}$ where $p$ is the number of DCT blocks of the original image. Each DCT block $X_i$ contains $8 \times 8$ quantised coefficients.

$X_i(j)$: the $j$th coefficient of DCT block $X_i$ along zig-zag scan order and $j$, the index of the DCT coefficients, is in the range of [0, 63] (see Figure 1(a)).

$X_i(h)$: the *explicitly* watermarkable coefficients with highest frequency $h$ in block $X_i$. Figure 1(b) highlights the four watermarkable coefficients in black background while the unwatermarkable non-zero coefficients are highlighted in gray background. In the example shown in Figure 1(b), $h$ is set to 19. Note since $X_i(17)$ is 0, it is not identified as watermarkable.

$A$: a binary sequence with the length of the same size as the image (i.e., $64 \times p$) generated with a secret key. It is organized in the same manner as $X$, e.g., $A_i(j)$ stands for the $j$th bit of block $A_i$ of $A$.

$B$: a binary sequence with the same length as that of $A$, which is also organized in the same manner as $X$. Any bit $B_i(j)$ is assigned a value of 1 if its corresponding DCT coefficient $X_i(j)$ is not zero, otherwise, 0 is assigned instead.

$W$: the binary watermark sequence resulted from the EXCLUSIVE-OR operation, denoted as $\oplus$, on $A$ and $B$. Therefore,

$$W = A \oplus B. \tag{1}$$

Like $B$, $W$ is also organized in the same manner as $X$ such that the $j$th bit of watermark block $i$ is denoted as $W_i(j)$. Since $B$ records the position of zero-valued DCT coefficients with '0', replacing a zero coefficient $X_i(j)$ with a non-zero value switches its corresponding bit $B_i(j)$ from 0 to 1, which in turn, messes up $W$ according to Eq (1).

$N_{9i}(j)$: the dependence neighbourhood / context of the watermarkable coefficient $X_i(j)$ comprising the 9 gray DCT blocks including $X_i$ itself as shown in Figure 2.

$S_i(j)$: the *secret sum* associated with watermarkable coefficient $X_i(j)$. It is the sum of the non-zero unwatermarkable coefficients within $N_{9i}$ selected according to their corresponding watermark bits and $W_i(j)$. It can be expressed as

$$S_i(j) = \sum_{m \in N_{9i}(j)} \sum_{n \in [0, h'-1]} (W_m(n) \oplus W_i(j)) \cdot X_m(n) \tag{2}$$

where $h'$ is the lowest frequency among the four watermarkable coefficients. Figure 3 shows how the coefficients in $X_i$ are selected for calculating $S_i(j)$. Suppose $X_i(16)$ in Figure 3(b) is to be watermarked. Since $W_i(16)$ is 0 as shown in black background in Figure 3(a), according the Eq (2), only the unwatermarkable non-zero coefficients (as highlighted in gray background in Figure 3(b)) with corresponding zero-valued watermark bits (as highlighted in gray background in Figure 3(a)) contributes to the calculation of $S_i(16)$. This idea applies to all the other 8 neighbouring blocks.

$T_i(j)$: the concatenation of $S_i(j)$ and $X_i(j)$ in two's complement format

### 3.1 Watermark embedding algorithm

Now, the proposed watermark embedding algorithm can be described as follows.

$Step_e$ 1. Perform DCT on the input image and quantise the DCT coefficients.

$Step_e$ 2. Generate $A$ with a secret key

$Step_e$ 3. Generate $B$

$Step_e$ 4. Generate the binary watermark $W$ according to Eq. (1)

$Step_e$ 5. For each DCT block $X_i$, repeat $Step_e$ 5.1 and $Step_e$ 5.2

$Step_e$ 5.1. Identify the four watermarkable coefficients

$Step_e$ 5.2. For each watermarkable coefficients $X_i(j)$, repeat $Step_e$ 5.2.1 and 5.2.2

$Step_e$ 5.2.1. Calculate $S_i(j)$ associated with $X_i(j)$ according to Eq. (2)

$Step_e$ 5.2.2. Modulate the selected coefficient $X_i(j)$ so that

$$Parity\ (T_i(j)) = W_i(j) \tag{3}$$

where *Parity* is a function which returns 1 or 0 as output to indicate that the number of '1' bits is *odd* or *even*

### 3.2 Authentication algorithm

For the verifier, the authentication algorithm works as follows:

$Step_a$ 1. Decode the received JPEG image to get the quantised DCT coefficients

$Step_a$ 2. Generate $A$ with a secret key

$Step_a$ 3. Generate $B$

$Step_a$ 4. Generate the binary watermark $W$ according to Eq. (1)

Step$_a$ 5. For each DCT block $X_i$, repeat Step$_a$ 5.1 and Step$_a$ 5.2

Step$_a$ 5.1. Identify the four watermarkable coefficients

Step$_a$ 5.2. For each watermarkable coefficients $X_i(j)$, repeat Step$_a$ 5.2.1 and Step$_a$ 5.2.2

Step$_a$ 5.2.1. Calculate $S_i(j)$ associated with $X_i(j)$ according to Eq. (2)

Step$_a$ 5.2.2. Authenticate the selected coefficients by verifying whether Eq. (3) holds or not. If the coefficient fails the authentication, i.e., Eq. (3) does not hold, the block, which the coefficient belongs to, is shaded to reduce the transparency so as to indicate the occurrence of tampering.

Step$_a$ 6. Turn off the false alarms. Any blocks marked as inauthentic surrounded by less than $k$ inauthentic blocks are treated as authentic. (Argued in Sec. 3.3).

### 3.3 Security analyses of the proposed algorithms

We can see, from the definition of $B$, that $B$ is intended for differentiating zero- and non-zero coefficients. It is possible to take the secret-key-generated binary sequence $A$ as the watermark $W$ without directly using $B$ and performing Eq. (1). However, without using $B$, attacking the zero coefficients creates spurious without altering their corresponding watermark bits. Those spurious coefficients can only be detected and involved as $X_m(n)$ in Eq. (2) when $W_m(n) \oplus W_i(j)$ equals 1. On the other hand, when $B$ is involved, manipulating the zero coefficients results in a different $B$ at the verifier's side. Consequently, according to Eq. (1), the watermarks $W$s used by the verifier and the embedder are different. In this case both the value of $X_m(n)$ and $W_m(n)$ in Eq. (2) are different. Therefore, the security is strengthened with the enforcement of $B$. It is now clear that the purpose of involving $B$ in Eq. (1) is to *register* and *implicitly watermark* the zero-valued coefficients, which will not

be *explicitly* watermarked.

Furthermore, by involving the secret sum $S_i(j)$ defined in Eq. (2) when watermarking coefficient $X_i(j)$, the non-zero unwatermarkable coefficients are also involved and is thus *implicitly watermarked* without distortion. These two features allow all the zero and non-zero unwatermarkable coefficients to be protected without being explicitly watermarked.

Barreto *et al*. [1] observed that, by using a nondeterministic signature, even the signatures of two identical images will be different. In our algorithm, the secret sum, $S_i$, serves the same purpose as the nondeterministic signature because for any block $i$ and their neighbouring blocks $m$, $m \in N_{9i}$, $W_i \neq W_m$. Therefore, even if two DCT blocks $X_i$ and $X_m$ are identical, their signatures $S_i$ and $S_m$ are still different. This feature makes the proposed algorithm immune to transplantation attack without resorting the relatively compute-intensive hashing operation.

Since the second-order neighbours are involved in the calculation of the secret sum $S_i$, any DCT block when manipulated will have direct impact on the correctness of the secret sum of its 8 second-order neighbours. That is to say that when a DCT block is manipulated, in addition to the *true* alarm raised by the block itself, it is highly likely that all its 8 neighbours will also raise *false* alarms depending on the values of the corresponding watermark bits of the manipulated coefficients. The false alarms reduce the accuracy of the tampering localization. $\text{Step}_a$ 6 of our authentication algorithm is meant for tackling this problem. Any blocks marked as inauthentic surrounded by less than $k$ ($0 \leq k \leq 7$) inauthentic blocks are treated as authentic. The smaller the value of $k$ is, the less forgiving the algorithm becomes, but the more inaccurate the algorithm is in locating tampering.

Figure 5 and *Experiment 1*demonstrate an example with $k = 6$.

## 4. EXPERIMENTS

In the following experiments, a decoded test image of size $248 \times 248$ pixels along with its watermarked version stamped with the proposed scheme are shown in Fig. 4(a) and 4(b), respectively. From Fig. 4(b) we can see that the distortion after adding the watermark is invisible. To test the effectiveness of our scheme, experiments are conducted by mounting local tampering, cropping, and low-pass filtering attacks on the watermarked image as follows. The value of $h$, the highest frequency among the four watermarkable coefficients is set to 12. The value of 6 is assigned to $k$ in $\text{Step}_a$ 6 of our authentication algorithm.

*Experiment 1. Local tampering:* The doorknob between the twin's head in the received image is removed, which involves only one DCT block, as shown in Fig. 5(a). The authentication result before $\text{Step}_a$ 6 is shown in Fig. 3(b), with the 9 blocks failing the authentication highlighted with shading effect. After $\text{Step}_a$ 6, the false alarms are turned off, with the only true alarm left on as shown in Fig. 5(c). The shaded block between the twins' head indicates that the received image has been locally tampered with.

*Experiment 2. Cropping:* The received images as shown in Fig. 6(a) is the cropped version of the watermarked image in Fig. 4(a). After authentication, the dominating shaded blocks in Fig. 6(b) indicate that the received image has been tampered with.

*Experiment 3. Low-pass filtering:* In order to test if the watermark can survive this attack, the received image is low-pass filtered as shown in Fig. 7(a). The dominating shaded blocks in Fig. 7(b) indicate that the proposed algorithm is capable of detecting this type of global image processing operations.

## 5. CONCLUSIONS

In this work, we pointed out that a secure fragile watermarking scheme must *implicitly* watermark *all* the coefficients and proposed a scheme based on this idea. The main contributions of the proposed scheme can be summarized as follows.

- By involving the unwatermarked non-zero coefficients in the watermarking process of one sixteenth of the DCT coefficients and registering the zero coefficients with the watermark, the proposed scheme minimises the distortion due to watermark embedding while providing the capability of authenticating all the coefficients including the zero ones.
- The simple idea of watermarking the DCT coefficients according to a secret sum extracted from a dependence neighbourhood puts up resistance to cropping, cover-up, vector quantization and transplantation attacks.
- The scheme is able to localize tampering such as removal of original objects and addition of foreign objects to the accuracy of individual DCT block *visually* and of individual DCT coefficient *mathematically*.
- High security and low computational complexity are achieved without using cryptography and hash function.
- Neither the original image nor other *a priori* knowledge is required in the watermark extraction process.

We are currently investigating the possibility of modifying this scheme so that it can be applied to the images compressed with JPEG 2000 standard, which involves wavelet transform.

## REFERENCES


[1] P. S. L. M. Barreto, H. Y. Kim, and V. Rijmen, “Toward secure public-key blockwise fragile authentication watermarking,” in *IEE Proceedings - Vision, Image and Signal Processing*, vol. 148, no. 2, pp. 57 – 62, April 2002.

[2] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, “A hierachical image authentication watermark with improved localization and security,” in *Proc. IEEE Int. Conf. Image Processing,* vol. II, Thessaloniki, Greece, October 2001, pp. 502-505.

[3] I. Cox, M. Miller, and B. Jeffrey, Digital Watermarking: Principles and Practice, Morgan Kaufmann, 2002.

[4] J. Fridrich, M. Goljan, and N. Memom, “Further attack on Yeung-Mintzer watermarking Scheme,” in *Proc. SPIE Electronic Imaging 2000*, San Jose, January 2000.

[5] J. Fridrich, M. Goljan, R Du, “Invertible Authentication Watermark for JPEG Images,” *IEEE Intl. Conf. on Information Technology,* Las Vegas, Nevada, April 2001, pp. 223–227.

[6] M. Holliman and N. Memon, “Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes,” *IEEE Trans. Image Processing,* vol. 9, no. 3, pp. 432-441, March 2000.

[7] T. Kalker and J. Haitsma, “Efficient detection of a spatial spread-spectrum watermark in MPEG vedio streams,” in *Proc. IEEE Int. Conf. Image Processing,* vol. I, Vancouver, Canada, Sept. 2000, pp. 434-437.

[8] C.-T. Li, D. C. Lou, and T. H. Chen, “Image Authenticity and Integrity Verification via Content-based Watermarks and a Public Key Cryptosystem,” in *Proc. IEEE Int. Conf. Image Processing,* vol. III, Vancouver, Canada, Sept. 2000, pp. 694-697.

[9] C.-T. Li and F.-M. Yang, "One-dimensional Neighbourhood Forming Strategy for Fragile Watermarking," *Journal of Electronic Imaging*, vol. 12, no 2, pp. 284-291, 2003.

[10] C.-T. Li, F. M. Yang, and C. S. Lee, "Oblivious Fragile Watermarking Scheme For Image Authentication," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol VI, Orlando, FL, USA, May 2002, pp 3445-3448.

[11] C. Y. Lin and S. F. Chang, “A robust image authentication method surviving JPEG lossy compression,” *Proc. SPIE*, vol. 3312, pp. 296-307, 1998.

[12] C. S. Lu, S. K. Huang, C. J. Sze, H. Y. Liao, “Cocktail watermarking for Digital Image Protection,” *IEEE Trans. Multimedia*, vol. 2, no. 4 December 2000.

[13]N. Memon, S. Shende, and P. W. Wong, “On the security of the Yeung-Mintzer Authentication Watermark,” in *Proc. IS & T PICS Symposium,* Savannah, Georgia, March 1999.

[14]W. Stallings, *Cryptography and network security – priciples and practice,* Prentice Hall, 1998.

[15]D. A. Winne, H. D. Knowles, D. R. Bull, and C. N. Canagarajah, "Digital Watermarking in Wavelet Domain with Predistortion for Authenticaticity Verification and Localization," in *Proceeding of SPIE: Security and Watermarking of Multimedia Contents IV,* vol. 4675, pp. 349-356, San Jose, January 2002.

[16]P. W. Wong, “A Public Key Watermark for Image Verification and Authentication,” in *Proc. IEEE Intl. Conf. Image Processing,* vol. I, Chicago, USA, October 1998, pp. 455-459.

[17]P. W. Wong and N. Memom, “Secret and public key authentication watermarking schemes that resist vector quantization attack,” in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, vol. 3971, no. 40, Jan. 2000.

[18]M. Wu and B. Liu, “Watermarking for Image Authentication, in *Proc. IEEE Intl. Conf. Image Processing,* vol. II, Chicago, USA, October 1998, pp. 437-441.

[19]L. Xie and G. R. Arce, “A class of authentication digital watermarks for secure multimedia communication,” *IEEE Trans. Image Processing,* vol. 10, no. 11, pp. 1754-1764, November 2001.

[20]M. Yeung and F. Minzter, “An Invisible Watermarking technique for image verification,” in *Proc. IEEE Intl. Conf. Image Processing,* vol. 1,Santa Barbara, USA, October 1997, pp. 680-683.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 5 | 6 | 14 | 15 | 27 | 28 |
| 2 | 4 | 7 | 13 | 16 | 26 | 29 | 42 |
| 3 | 8 | 12 | 17 | 25 | 30 | 41 | 43 |
| 9 | 11 | 18 | 24 | 31 | 40 | 44 | 53 |
| 10 | 19 | 23 | 32 | 39 | 45 | 52 | 54 |
| 20 | 22 | 33 | 38 | 46 | 51 | 55 | 60 |
| 21 | 34 | 37 | 47 | 50 | 56 | 59 | 61 |
| 35 | 36 | 48 | 49 | 57 | 58 | 62 | 63 |

(a)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 17 | 14 | -9 | 8 | -2 | 2 | 0 | 0 |
| -1 | 11 | 10 | -1 | 2 | 0 | 0 | 0 |
| -1 | 9 | 3 | 0 | 0 | 0 | -1 | 0 |
| 6 | -4 | -1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(b)

Figure 1. (a) Indices of DCT coefficients in zig-zag scan order. (b) Quantised DCT coefficient block $X_i$. The coefficients in the black background are identified as *watermarkable* whereas the ones in gray background are *nonwatermarkable*. In this example, the value of $h$, the highest frequency among the four watermarkable coefficients, is set to 19. Since $X_i(17)$ is zero, it is not taken as watermarkable.

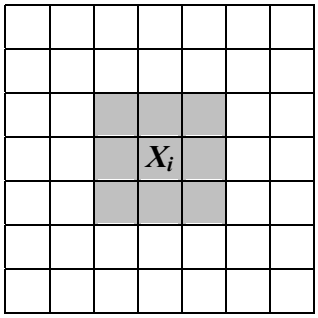


Figure 2. The 9-neighbourhood systems, the gray region, of DCT block $\boldsymbol{X_i}$. The neighbourhood comprises the traditional second-order neighbourhood of $\boldsymbol{X_i}$ and itself.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

(a)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 17 | 14 | -9 | 8 | -2 | 2 | 0 | 0 |
| -10 | 11 | 10 | -1 | 2 | 0 | 0 | 0 |
| -12 | 9 | 3 | 0 | 0 | 0 | -1 | 0 |
| 6 | -4 | -1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(b)

Figure 3. (a) A watermark block $W_i$. (b) The DCT coefficient block $X_i$ same as Figure 1(b). Suppose $X_i(16)$ is to be watermarked, since $W_i(16)$ is 0 as shown in black background in (a), according the Eq (2), only the unwatermarkable non-zero coefficients (as shown in gray background in (b)) with corresponding zero-valued watermark bits (as highlighted in gray background in (a)) contributes to the calculation of $S_i(16)$.

(a) (b)

Figure 4. (a) De-compressed original image. (b) De-compressed watermarked image.

(a)

(b) (c)

Figure 5. (a) Received image. The doorknob between the twins' heads has been masked. (b) Authentication result before the false alarms are turned off. (c) Authentication result after the false alarms are turned off. The shaded block between the twins' head indicates that the received image has been locally tampered with.

(a) (b)

Figure 6. (a) Received image – a cropped version of the watermarked image. (b) Authentication result. The dominating shaded blocks indicate that the received image has been tampered with.

(a) (b)

Figure 7. (a) Received image – a low-pass filtered version of the watermarked image. (b) Authentication result. The dominating shaded blocks indicate that the received image has been tampered with.